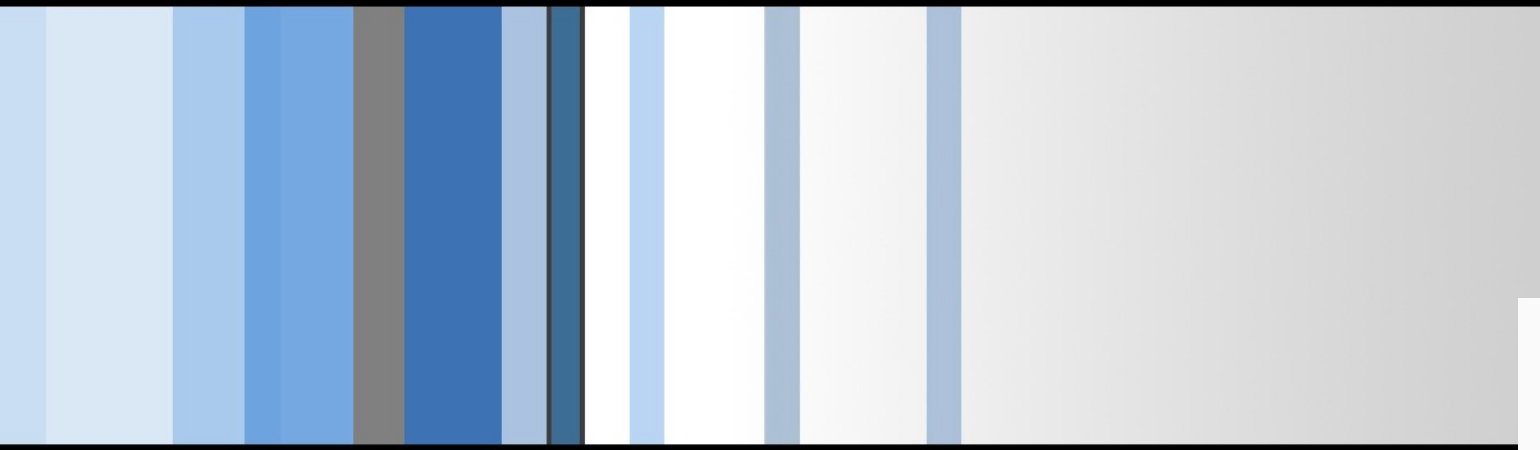
Class 18

CSC 496: iOS App Development
SpriteKit

Si Chen (schen@wcupa.edu)



SpriteKit

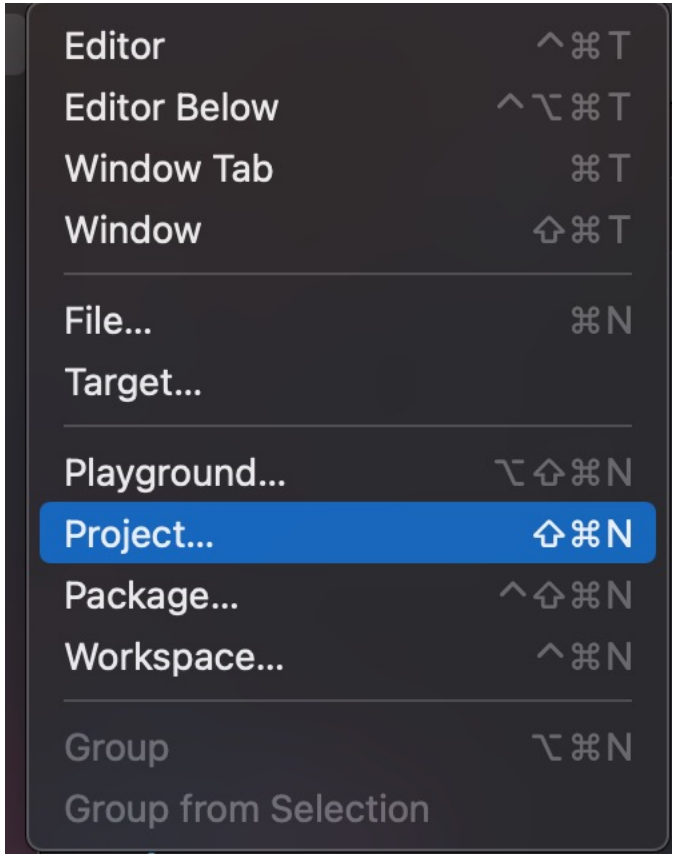
Framework

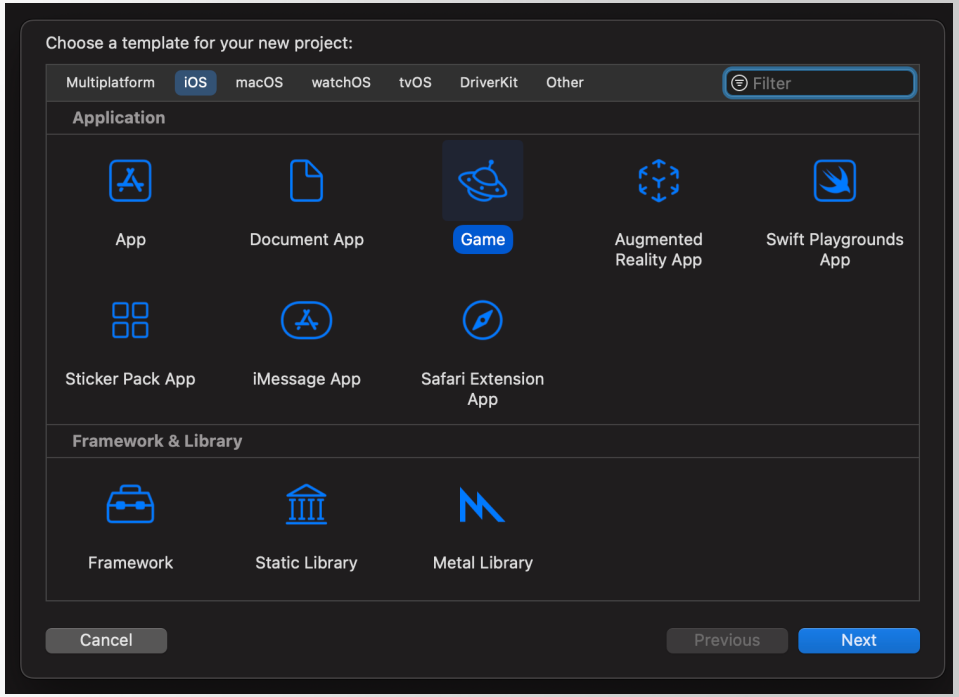
SpriteKit

Add high-performance 2D content with smooth animations to your app, or create a game with a high-level set of 2D game-based tools.

SpriteKit

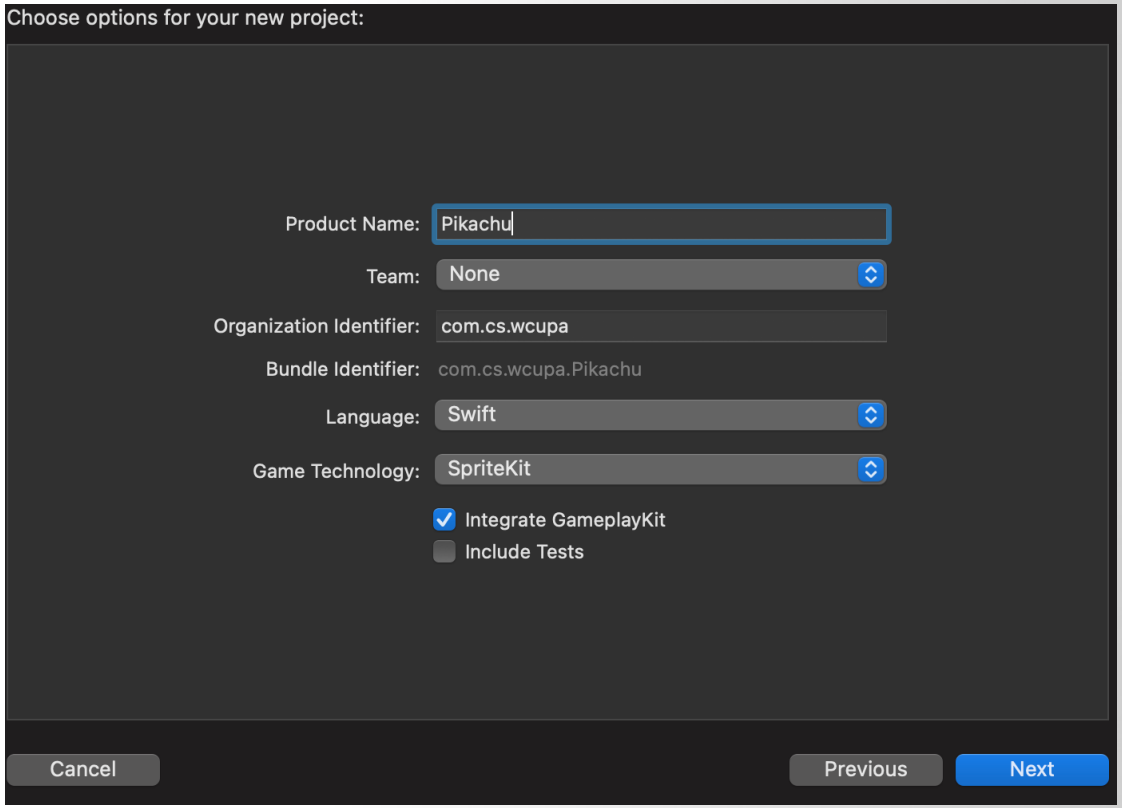
Creating Scenes with Sprites and Nodes





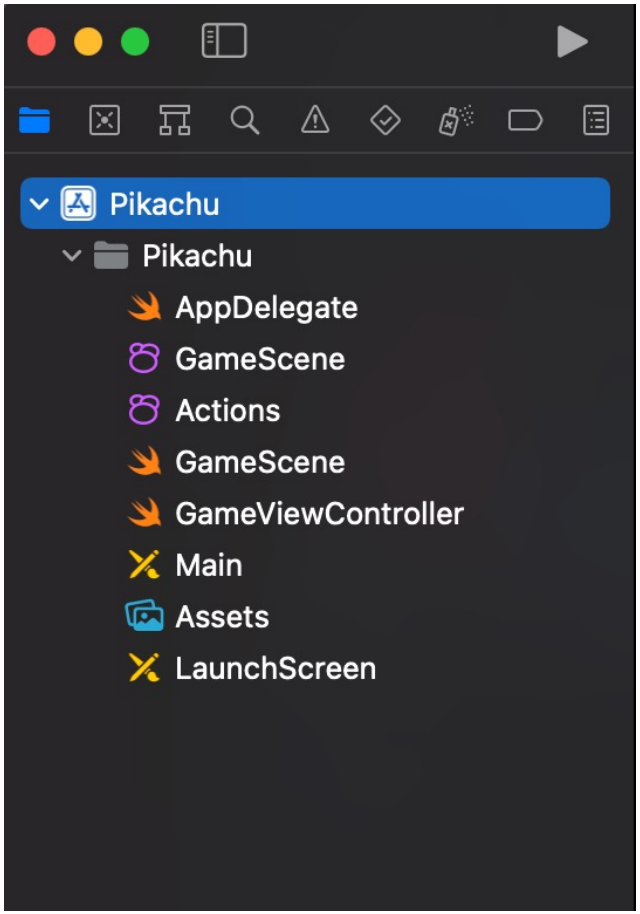
Start up Xcode, select **File\New\Project…**, choose the **iOS\Game** template and click **Next**.

Creating Scenes with Sprites and Nodes



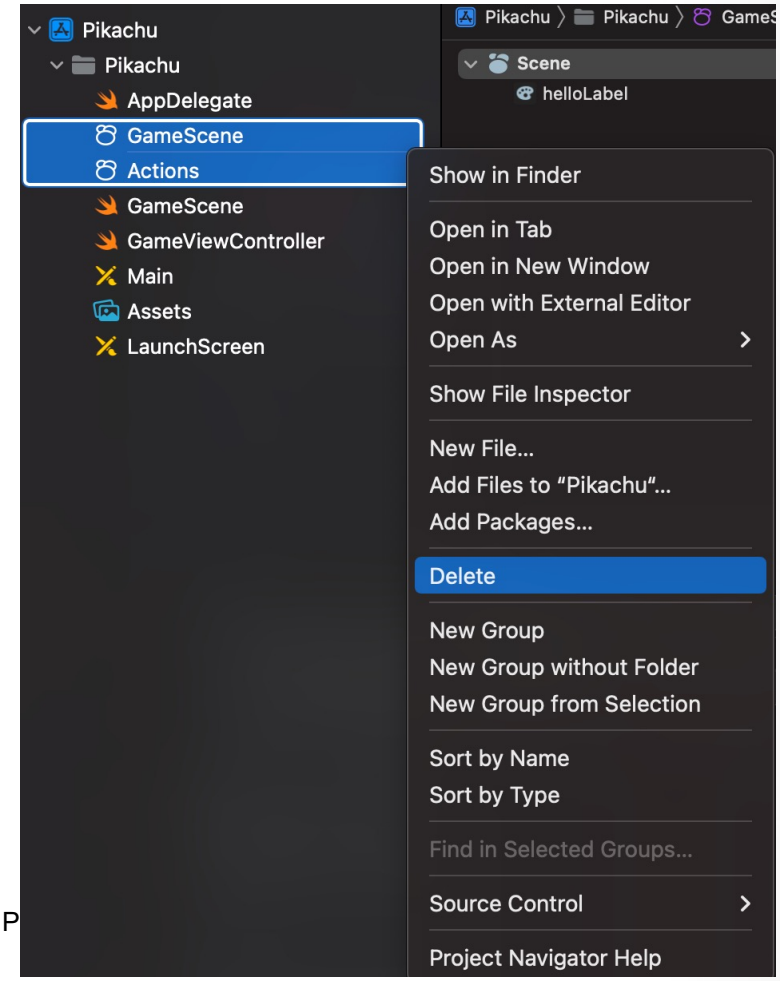
- Enter **Pikachu** for the Product Name, **Swift** for Language, **SpriteKit** for Game Technology. Make sure the option for Integrate GameplayKit is checked, and the option for Include Tests is **unchecked** and click **Next**:

Creating Scenes with Sprites and Nodes



Clean Up the Default Template

- The default iOS game template includes a lot of boilerplate code and files you won't need, so it's best to remove these things.



Delete **GameScene.sks**, and **Actions.sks**

Clean Up the Default Template

- Delete everything in GameScene.swift and replace it with this:

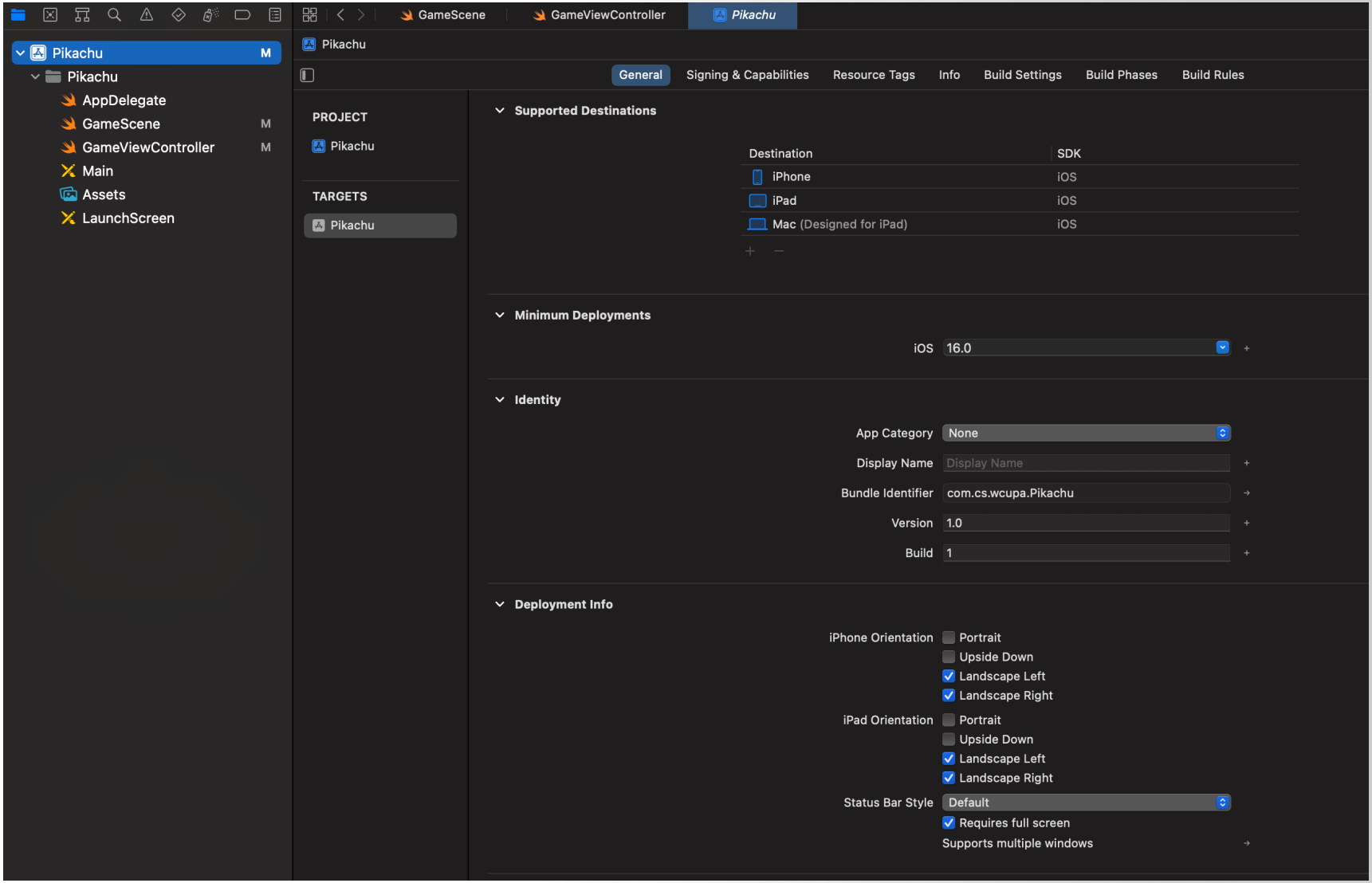
```
8   import SpriteKit
9   import GameplayKit
10
11  class GameScene: SKScene {
12
13      override func didMove(to view: SKView) {
14
15      }
16
17  }
```

Clean Up the Default Template

- Open **GameViewController.swift** and remove all of the code inside the ***viewDidLoad()*** method, leaving only the line that reads ***super.viewDidLoad()***

```
8  import UIKit
9  import SpriteKit
10 import GameplayKit
11
12 class GameViewController: UIViewController {
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         
17     }
18
19     override var supportedInterfaceOrientations: UIInterfaceOrientationMask {
20         if UIDevice.current.userInterfaceIdiom == .phone {
21             return .allButUpsideDown
22         } else {
23             return .all
24         }
25     }
26
27     override var prefersStatusBarHidden: Bool {
28         return true
29     }
30 }
```

Set the Supported Device Orientation



Disabling the ***Portrait*** option and select "***Requires full scene***"

Set the Supported Device Orientation

In the GameViewController.swift

```swift
import UIKit
import SpriteKit
import GameplayKit

class GameViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

    }

    override var supportedInterfaceOrientations: UIInterfaceOrientationMask {
        return .landscape
    }

    override var prefersStatusBarHidden: Bool {
        return true
    }
}
```

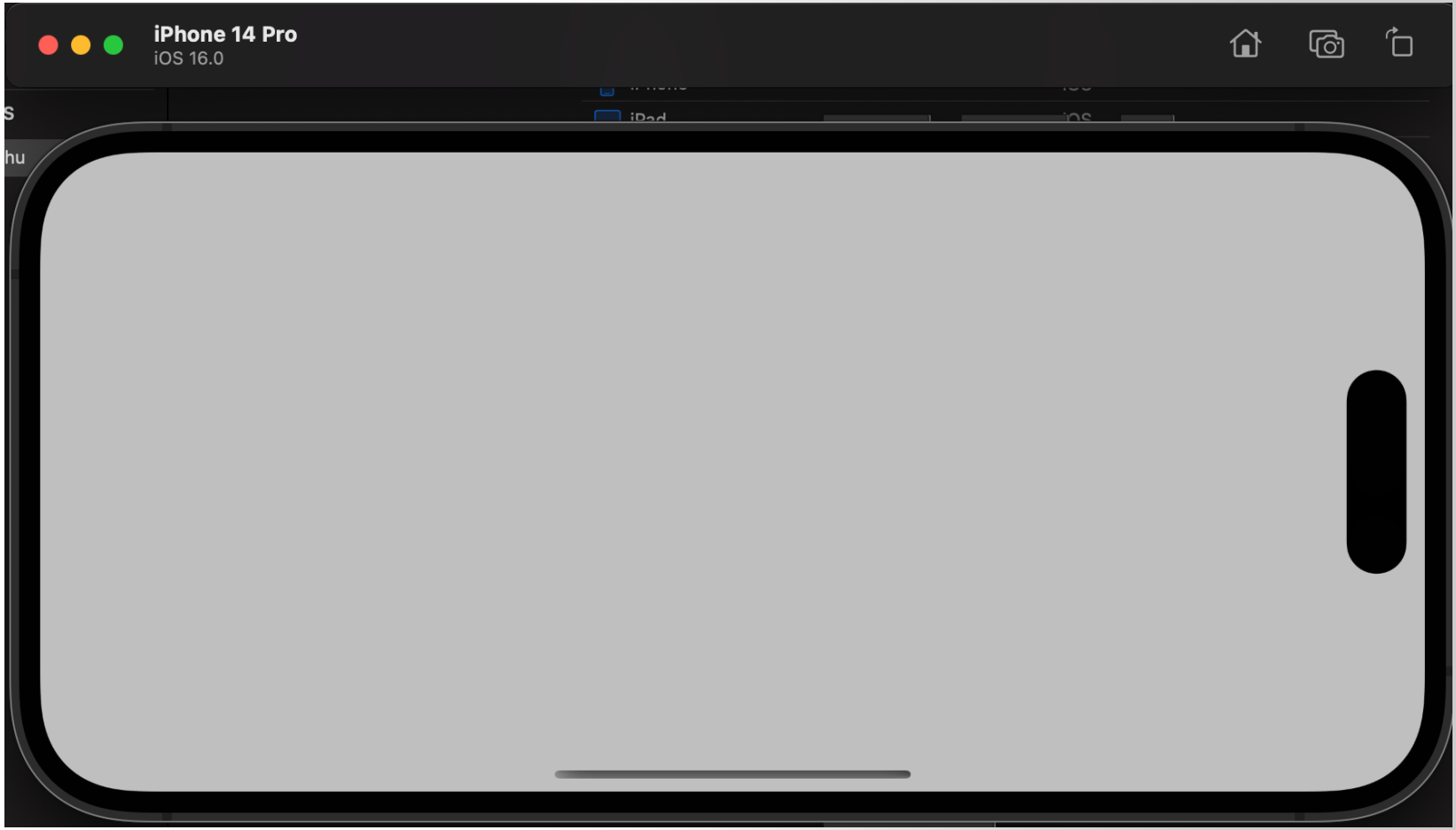
Modify

```swift
override var supportedInterfaceOrientations: UIInterfaceOrientationMask {
    if UIDevice.current.userInterfaceIdiom == .phone {
        return .allButUpsideDown
    } else {
        return .all
    }
}
```

to

```swift
override var supportedInterfaceOrientations: UIInterfaceOrientationMask {
    return .landscape
}
```

An Empty Game



Create Your First SpriteKit Scene

- Some key components that make up a SpriteKit scene :
  - SKView: This is the primary view for a SpriteKit scene. The SKView class inherits from the UIView class
  - SKScene: This is the root node of the scene. The SKScene class includes properties and methods that define how to render content and process animation.

- Let’s create our first scene

Create Your First SpriteKit Scene

```
override func viewDidLoad() {
    super.viewDidLoad()
    // create the view
    if let view = self.view as! SKView?{
        // create the scene
        let scene = GameScene(size: CGSize(width: 1336, height: 1024))

        // set the scale mode to scale to fill the view window
        scene.scaleMode = .aspectFill

        // set the background color
        scene.backgroundColor = UIColor(red: 105/255, green: 157/255, blue: 181/255, alpha: 1.0)

        // present the scene
        view.presentScene(scene)

        // set the view options
        view.ignoresSiblingOrder = false
        view.showsPhysics = false
        view.showsFPS = true
        view.showsNodeCount = true

    }
}
```

Create Your First Sprite Node

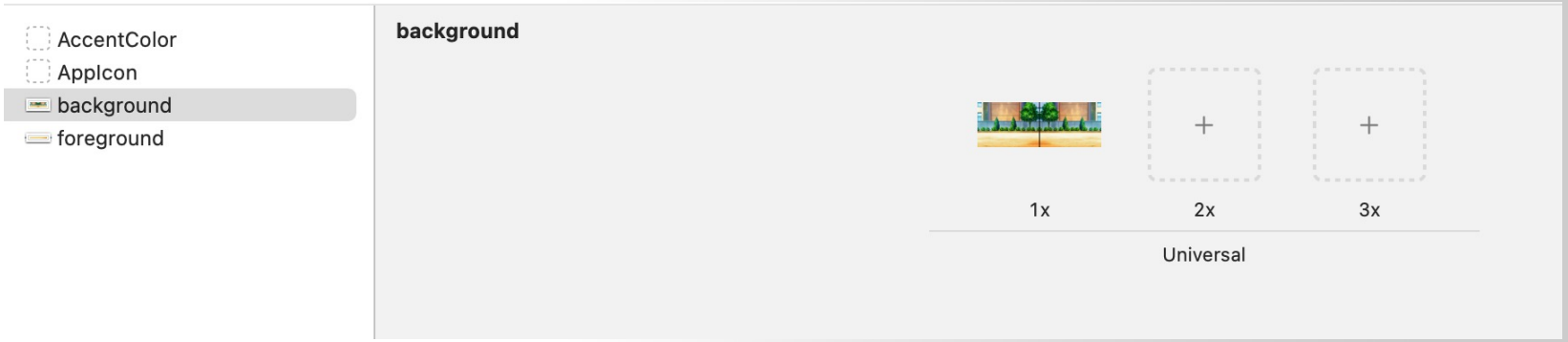
- The SKNode class doesn't render (draw) any visual content – it is considered the building block of SpriteKit because every node in a SpriteKit scene is a subclass of SKNode.
- To draw content, you need to use a visual node, such as:
  - SKSpriteNode: this node draws a rectangle texture, image or color.
  - SKShapeNode: Used along with a core graphics path to draw custom shapes.
  - SKLableNode: When you need text, this type of node is used to draw a text label
  - SKVideoNode: Display video content
  - SKReferenceNode: create reusable content

Create Your First Sprite Node

- Although the SKNode class does not allow for visual content, it does provide some standard properties that its subclasses inherit.
  - Position:
    - Frame
    - Position
    - zPosition
  - Scale and Rotation:
    - xScale
    - yScale
    - zRotation

Add Image Assets

- Please download and add (draw and drop) background and foreground image to your project's Assets



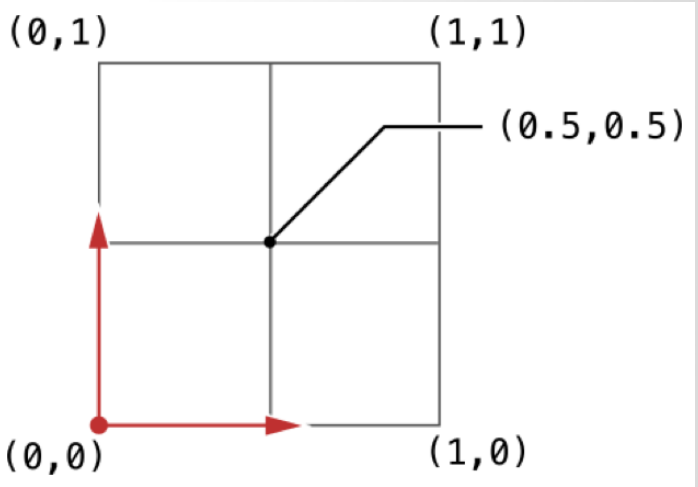
Add the Background

Inside GameScene.swift, add the following code

```
import SpriteKit
import GameplayKit

class GameScene: SKScene {

    override func didMove(to view: SKView) {
        let background = SKSpriteNode(imageNamed: "background")
        background.anchorPoint = CGPoint(x:0, y:0)
        background.position = CGPoint(x: 10, y: 330)
        addChild(background)

    }

}
```

Anchor Point

- A sprite node's anchorPoint determines which point within its frame corresponds to its position.
- Anchor points are specified in the unit coordinate system, shown in the following illustration.
- The unit coordinate system places the origin at the bottom left corner of the frame and (1,1) at the top right corner of the frame. A sprite's anchor point defaults to (0.5,0.5), which corresponds to the center of the frame.

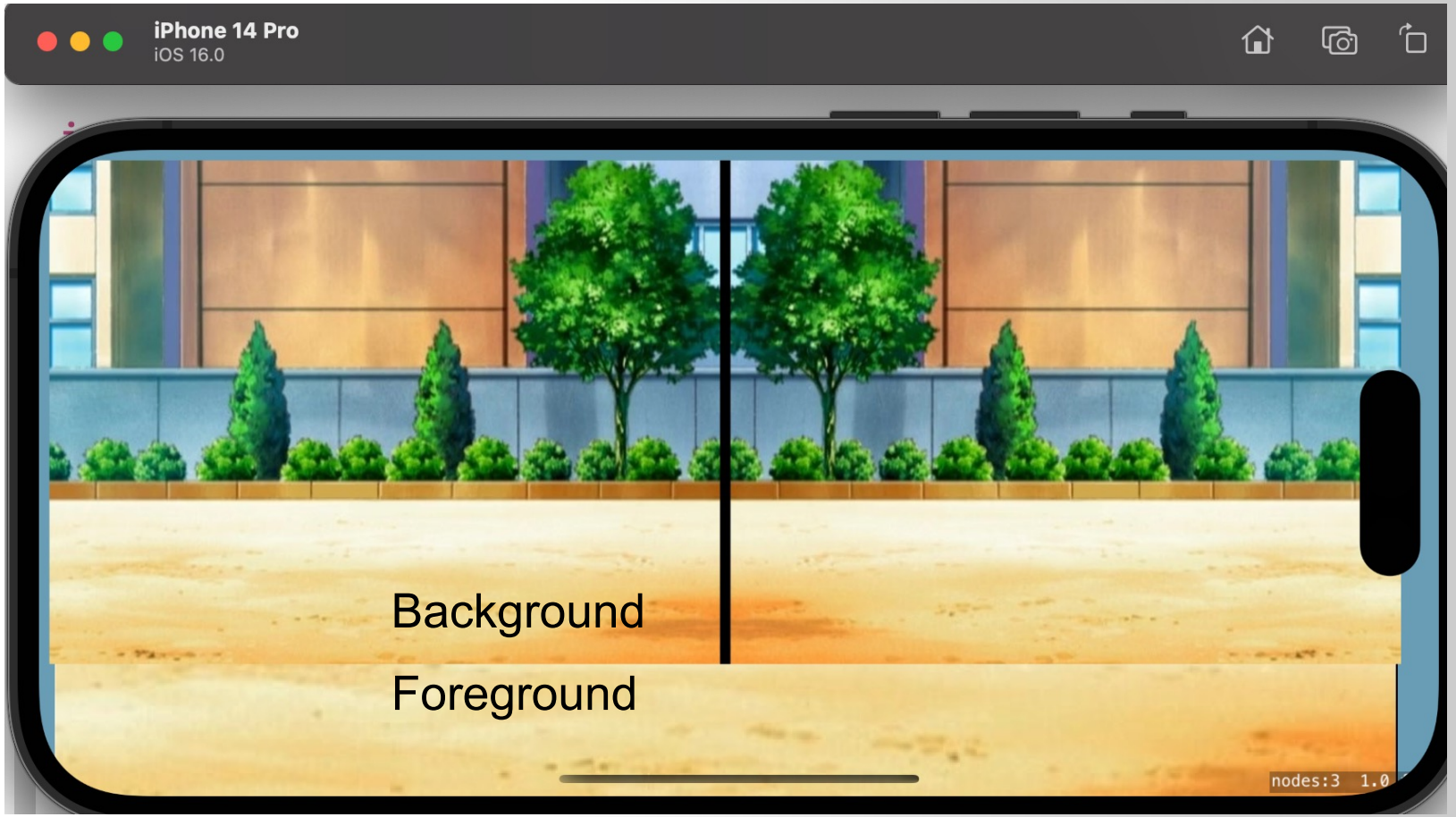



```
background.anchorPoint = CGPoint(x:0, y:0)
```

Add the Foreground

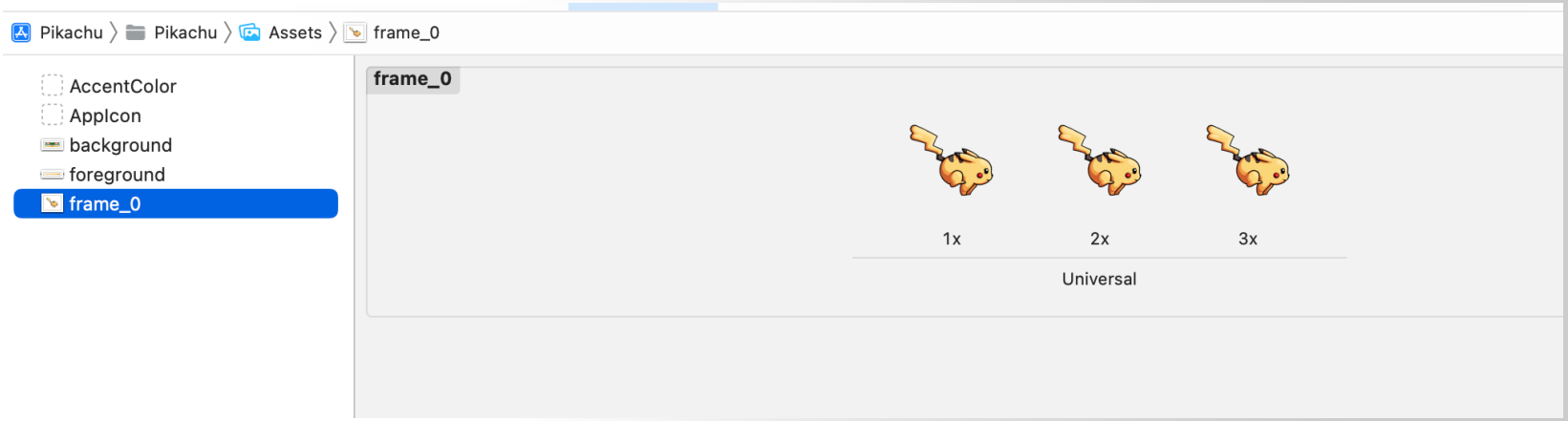
Inside GameScene.swift, add the following code

```
class GameScene: SKScene {

    override func didMove(to view: SKView) {
        let background = SKSpriteNode(imageNamed: "background")
        background.anchorPoint = CGPoint(x:0, y:0)
        background.position = CGPoint(x: 10, y: 330)
        addChild(background)

        let foreground = SKSpriteNode(imageNamed: "foreground")
        foreground.anchorPoint = CGPoint(x: 0, y: 0)
        foreground.position = CGPoint(x:15, y:158)
        addChild(foreground)
    }

}
```


iPhone 14 Pro
iOS 16.0
Background
Foreground
nodes:3 1.0

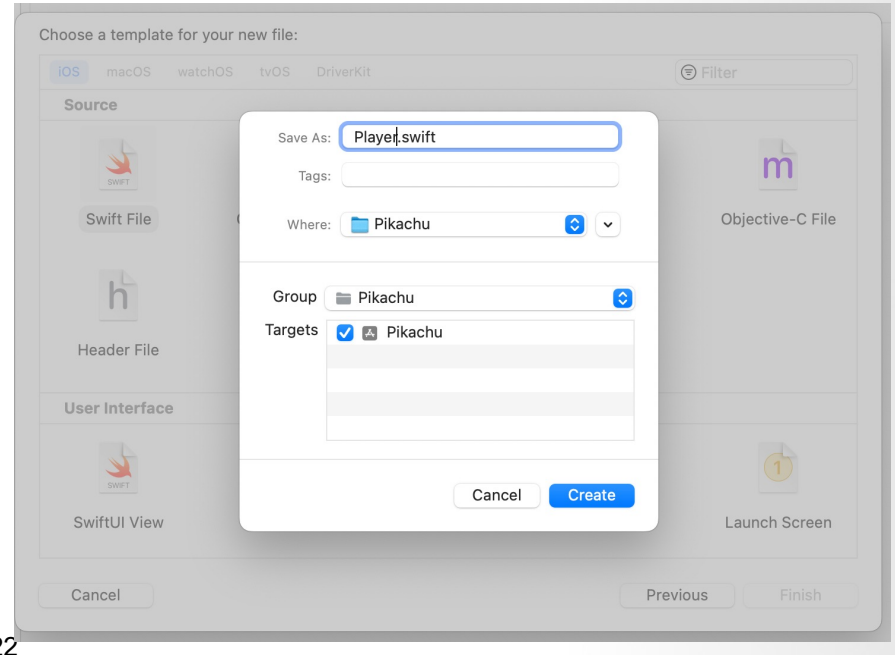
Add player - Pikachu

- Download and unzip the Pikachu (frame0) from d2L and add it (drag and drop) to your project's asset.



Create Player Class

- While it's possible to add all of your code to the GameScene Class. It's usually not the best plan and can create programs further down the road when you want to extend your game's features.
- Instead, you'll create a separate Player class to keep your codebase free of the muck that causes spaghetti code.



Create Player Class

```
import Foundation
import SpriteKit

class Player:SKSpriteNode{
    // MARK: - PROPERTIES

    // MARK: - INIT
}
```

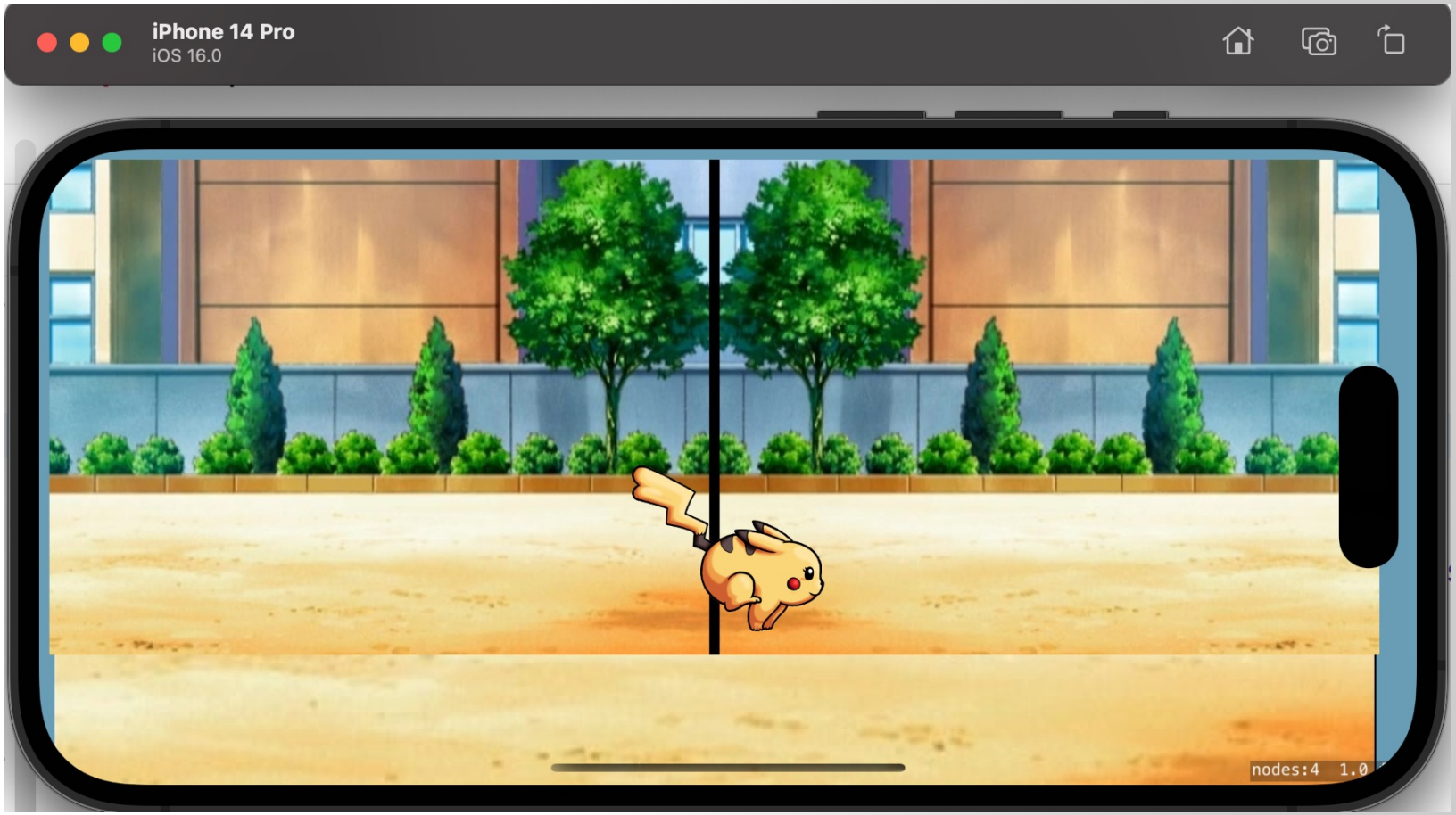
Create Player Class

```
import Foundation
import SpriteKit

class Player:SKSpriteNode{
    // MARK: - PROPERTIES

    // MARK: - INIT
    init(){
        // set default texture
        let texture = SKTexture(imageNamed: "frame_0")

        // call to super.init
        super.init(texture: texture, color: .clear, size: texture.size())

        self.name = "player"
        self.setScale(1.0)
        self.anchorPoint = CGPoint(x: 0.5, y:0.0)

    }
    required init?(coder aDecoder: NSCoder){
        fatalError("init(coder:) has not been implemented")
    }

}
```

Add player to scene

```
class GameScene: SKScene {

    override func didMove(to view: SKView) {
        let background = SKSpriteNode(imageNamed: "background")
        background.anchorPoint = CGPoint(x:0, y:0)
        background.position = CGPoint(x: 10, y: 330)
        background.zPosition = Layer.background.rawValue
        addChild(background)

        let foreground = SKSpriteNode(imageNamed: "foreground")
        foreground.anchorPoint = CGPoint(x: 0, y: 0)
        foreground.position = CGPoint(x:15, y:158)
        foreground.zPosition = Layer.foreground.rawValue
        addChild(foreground)

        let player = Player()
        player.position = CGPoint(x: size.width/2, y: foreground.frame.maxY)
        addChild(player)
    }

}
```

Run our game



Create SpriteKitHelper.Swift class to set Z Position

```
import Foundation
import SpriteKit

enum Layer:CGFloat{
    case background
    case foreground
    case player
}
```

Create SpriteKitHelper.Swift class to set Z Position

- In GameScene.Swift

```
import SpriteKit
import GameplayKit

class GameScene: SKScene {

    override func didMove(to view: SKView) {
        let background = SKSpriteNode(imageNamed: "background")
        background.anchorPoint = CGPoint(x:0, y:0)
        background.position = CGPoint(x: 10, y: 330)
        background.zPosition = Layer.background.rawValue
        addChild(background)

        let foreground = SKSpriteNode(imageNamed: "foreground")
        foreground.anchorPoint = CGPoint(x: 0, y: 0)
        foreground.position = CGPoint(x:15, y:158)
        foreground.zPosition = Layer.foreground.rawValue
        addChild(foreground)

        let player = Player()
        player.position = CGPoint(x: size.width/2, y: foreground.frame.maxY)
        addChild(player)
    }

}
```

Create SpriteKitHelper.Swift class to set Z Position

- In Player.Swift

```
import Foundation
import SpriteKit

class Player:SKSpriteNode{
    // MARK: - PROPERTIES

    // MARK: - INIT
    init(){
        // set default texture
        let texture = SKTexture(imageNamed: "frame_0")

        // call to super.init
        super.init(texture: texture, color: .clear, size: texture.size())

        self.name = "player"
        self.setScale(1.0)
        self.anchorPoint = CGPoint(x: 0.5, y:0.0)
        self.zPosition = Layer.player.rawValue
    }
    required init?(coder aDecoder: NSCoder){
        fatalError("init(coder:) has not been implemented")
    }

}
```

- Add player movement

Player.swift

```
import Foundation
import SpriteKit

class Player:SKSpriteNode{
    // MARK: - PROPERTIES

    // MARK: - INIT
    init(){
        // set default texture
        let texture = SKTexture(imageNamed: "frame_0")

        // call to super.init
        super.init(texture: texture, color: .clear, size: texture.size())

        self.name = "player"
        self.setScale(1.0)
        self.anchorPoint = CGPoint(x: 0.5, y:0.0)
        self.zPosition = Layer.player.rawValue
    }
    required init?(coder aDecoder: NSCoder){
        fatalError("init(coder:) has not been implemented")
    }

    func moveToPosition(pos: CGPoint, speed: TimeInterval){
        let moveAction = SKAction.move(to: pos, duration: speed)
        run(moveAction)
    }
}
```

GameScene.Swift

```swift
import SpriteKit
import GameplayKit

class GameScene: SKScene {
    let player = Player()

    override func didMove(to view: SKView) {
        let background = SKSpriteNode(imageNamed: "background")
        background.anchorPoint = CGPoint(x:0, y:0)
        background.position = CGPoint(x: 10, y: 330)
        background.zPosition = Layer.background.rawValue
        addChild(background)

        let foreground = SKSpriteNode(imageNamed: "foreground")
        foreground.anchorPoint = CGPoint(x: 0, y: 0)
        foreground.position = CGPoint(x:15, y:158)
        foreground.zPosition = Layer.foreground.rawValue
        addChild(foreground)

        player.position = CGPoint(x: size.width/2, y: foreground.frame.maxY)
        addChild(player)
    }

    func touchDown(atPoint pos: CGPoint){
        player.moveToPosition(pos: pos, speed: 1.0)
    }

    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        for t in touches{self.touchDown(atPoint: t.location(in: self))}
    }
}
```

Q & A